

REMARKS

The applicants request reconsideration of the rejection.

Claims 1-12 are now pending.

Claims 1-10 were rejected under 35 U.S.C. §103(a) as being unpatentable over Ford, US 5,497,457 (Ford) in view of Tomita, US 2003/0115437 (Tomita). The Applicants traverse as follows.

The Office Action alleges that Ford supports the rejection as a primary reference, citing primarily Fig. 2 and the associated disclosure. However, Ford's tape library access system does not appear to have the fundamental structural features from which the present invention advances the art, let alone the advancements themselves.

At the outset, the Applicants note that Ford is directed to a "redundant array of independent data storage libraries of dismountable media units", for which prior art problems of inefficient access are addressed. Specifically, whereas the prior art suffered from problems of access delay caused by the long time required to dismount unneeded tapes (or other media) and mount tapes having the desired storage locations or stored data, Ford discloses a system whereby parity updates are computed for new data and stored in a parity log in separate nonvolatile storage. The new data blocks are written onto

mounted media without writing the corresponding new parity if the media unit containing the corresponding parity block to be updated is not mounted. Then, when parity updates for a plurality of new data blocks have accumulated, the media unit containing the parity blocks to be updated is mounted and the parity updates retrieved for writing therein.

Turning to claim 1 as representative (all independent claims contain similar language), serial data is read from a storage device (such as cache memory 206), to which received data is to be written, in block units and stored in a data storage unit (such as data storage unit 504). When a block of data in a storage area of the storage device is the same as the serial data read from the storage area of the storage device and stored in the data storage unit, the serial data stored in the data storage unit is updated by means of the write data.

On the other hand, when a block of data in the storage area of the storage device is different from the serial data read from the storage area of the storage device and stored in the data storage unit, a security code is generated based on the serial data stored in the data storage unit, and the generated security code is added to the serial data stored in the data storage unit and transferred to the storage device.

Further the serial data stored in the block in the storage area of the storage device constituting the write destination of the write data is read, and the data stored in the data storage unit is updated by means of the write data.

The Office Action asserts correspondence between the data control unit of the claim (which performs the stated functions) and Ford's SCSI bus controller 510. However, Ford's SCSI bus controller 510 does not read serial data from a storage device in block units and store the data in random access buffers 508 (asserted as the data storage unit). In the passage noted by the Examiner (Col. 5, lines 13-14), when a data block is to be written to the library system, the new data block is accepted and stored in a buffer, but the data is not said to be serial data, read from a storage device to which the received data is to be written, in block units as claimed.

Moreover, Ford does not teach that, when a block of data in a storage area of a storage device (to which the received data is to be written) is the same as the serial data read from the storage area of the storage device and stored in the random access buffers, the serial data stored in the data storage unit is updated by means of the write data. The passage cited against this limitation (Col. 2, lines 38-44)

simply states that when new data blocks are to be written, the parity updates are computed and stored in a parity log, and the new data blocks are then written into the media units without writing the corresponding new parity block if the media unit containing the corresponding parity block to be updated is not mounted. There is no comparison of data blocks taken from the storage device as required by the claim, as a condition for updating the data in the random access buffers.

Further, Ford does not suggest that, when a block of data in the storage area of the storage device is different from the serial data read from the storage area of the storage device and stored in the random access buffers, a security code is generated based on the serial data stored in the random access buffers, and the generated security code is added to the serial data stored in the random access buffers and transferred to the storage device. Ford does not therefore suggest that the serial data stored in the block in the storage area of the storage device constituting the write destination of the write data is read, and the data stored in the random access buffers is updated by means of the write data. The passages noted in the Office Action (Col. 2, lines 38-44; Col. 4, lines 43-48; and Col. 5, lines 1-3) refer to the separate updating of data and parity discussed above, and

to RAID 5 parity generation in general. They do not address the situation described in the claim, in which a comparison of the data blocks is a condition for adding a security code to the data before transferring it to the storage device.

The secondary reference to Tomita is cited as disclosing that the "storage-drive data is the storage drive location." However, this quoted language is not a feature of the original or amended claims. Further, Tomita does not disclose or fairly suggest the limitations missing from Ford, as discussed above.

In view of the foregoing amendments and remarks, the Applicants request reconsideration of the rejection and allowance of the claims.

Respectfully submitted,


Daniel J. Stanger
Registration No. 32,846
Attorney for Applicants

MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C.
1800 Diagonal Road, Suite 370
Alexandria, Virginia 22314
(703) 684-1120
Date: November 8, 2005